2

AMENDMENTS TO THE CLAIMS

1 - 3. (Cancelled)

4. (Currently amended) A method according to Claim ~~2~~ 20 wherein each message data structure includes a sender queue and a receiver queue, for queuing a ~~number~~ plurality of sender and receiver processes.

5. (Currently amended) A method according to Claim ~~2~~ 20 further including checking the message data structure when a message has been processed, to determine whether there is at least one remaining sender and receiver in the message data structure for the message and, if so, rescheduling the message.

6. (Currently amended) A method according to Claim ~~2~~ 20 wherein each message data structure holds pointers to a composition activity, for composing a higher-level message from a lower-level message, and to a decomposition activity, for decomposing a higher-level message into a lower-level message.

7 - 8. (Cancelled)

9. (Currently amended) A method according to Claim 6 wherein:
- the decomposition activity for a message is activated when a process is added to the message data structure as a sender for that message; and
- the composition activity for a message is activated when a process is added to the message data structure as a receiver for that message.

10. (Currently amended) A method according to Claim ~~1~~ 20 wherein the step of scheduling the messages for processing comprises providing at least one scheduler queue, which is used for scheduling both messages and processes.

11. (Original) A method according to Claim 10 including the steps:
- scheduling the processes and messages by placing process-type items and message-type items on the scheduler queue;

3

- processing each process-type item on the scheduler queue by calling the process to which the item relates; and
- processing each message-type item on the scheduler queue by calling both the sender and receiver processes of the message to which the item relates.

12 -17. (Cancelled)

18. (Currently amended) A computer-implemented simulation method comprising the steps:
- modelling a target system as a set of processes that communicate with each other by way of messages;
- associating the messages with sender and receiver processes;
- providing at least one scheduler queue, holding a series of items, each item having a type value which indicates the item type as being either a process-type item or a message-type item;
- scheduling the processes by placing process-type items on the scheduler queue and scheduling the messages by placing process-type items and message-type items on the scheduler queue;
- processing each process-type item on the scheduler queue by calling the process to which the item relates; and
- processing each message-type item on the scheduler queue by calling both the sender and receiver processes of the message to which the item relates.

19. (Cancelled)

20. (New) A computer-implemented simulation method comprising:
(a) modelling a target system as a set of processes that communicate with each other by way of messages;
(b) associating a message data structure with each of the messages;
(c) when a process requires to send a message, adding that process to the relevant message data structure as a sender process;
(d) when a process requires to receive a message, adding that process to the relevant message data structure as a receiver process;

4

(e) scheduling a message for processing when there is at least one sender process and at least one receiver process in the message structure associated with the message; and

(f) processing each scheduled message by calling the sender and receiver processes in the message data structure associated with the message.

21. (New) A data carrier, carrying a computer-readable program for performing a computer-implemented simulation method comprising:

(a) modelling a target system as a set of processes that communicate with each other by way of messages;

(b) associating a message data structure with each of the messages;

(c) when a process requires to send a message, adding that process to the relevant message data structure as a sender process;

(d) when a process requires to receive a message, adding that process to the relevant message data structure as a receiver process;

(e) scheduling a message for processing when there is at least one sender process and at least one receiver process in the message structure associated with the message; and

(f) processing each scheduled message by calling the sender and receiver processes in the message data structure associated with the message.